

```

set nocompatible filetype off

set rtp+=~/vim/bundle/Vundle.vim call vundle#begin('~/.config/nvim/bundle')

" This should be propagated onto the laptop

Plugin 'VundleVim/Vundle.vim' Plugin 'valloric/youcompleteme' Plugin 'rdnetto/YCM-
Generator' Plugin 'ervandew/supertab'

Plugin 'itchyny/lightline.vim' "Plugin 'bling/vim-airline'"Plugin 'itchyny/lightline.vim'
"Plugin 'powerline/powerline',{rtp:'powerline/bindings/vim/'} Plugin
'NLKNguyen/papercolor-theme' Plugin 'tikhomirov/vim-gsl' Plugin 'hdima/python-syntax'
Plugin 'plasticboy/vim-markdown'

Plugin 'scrooloose/nerdtree'

Plugin 'tpope/vim-surround' Plugin 'jiangmiao/auto-pairs' Plugin 'sirver/ultisnips'

Plugin 'lervag/vimtex' "Plugin 'honza/vim-snippets' Plugin 'tpope/vim-obsession'

call vundle#end() filetype plugin indent on

" " Options " set encoding=utf-8

"line numbers set number set relativenumber

"tabs set softtabstop=-1 set noexpandtab set tabstop=4 set shiftwidth=4

"visual tab stuff set list set listchars=tab: | ,extends:»,precedes:«,nbsp:,trail:

"cursor line set nocursorline

"tab and status line set showtabline=2" Always display the tabline, even if there is only one
tab set laststatus=2

"wraps and indentation set nowrap set autoindent set smartindent

"color and syntax syntax enable set t_Co=256 set background=dark colorscheme
PaperColor

"terminal mouse default mode set mouse=a

set clipboard=unnamedplus

"GUI configuration set guioptions-=T set guioptions-=r set guioptions-=L set guioptions-=e
set guifont=Sudo:h14.5

set completeopt=preview,menuone

```

```

" "folding set foldmethod=syntax set foldenable set fillchars=fold: set
foldtext=MyFoldText() highlight Folded ctermbg=black ctermfg=grey cterm=italic highlight
Folded guibg=black guifg=grey gui=italic

function MyFoldText() let line = getline(v:foldstart) let num = (v:foldend - v:foldstart) return
[" . num . "]" . " " . line endfunction

"vsplit"set fillchars+=vert: "highlight VertSplit guifg=grey

" " Highlights " highlight Comment cterm=italic gui=italic highlight SpecialKey NONE
highlight CursorLine cterm=NONE ctermbg=Black guibg=DarkGrey highlight ColorColumn
ctermbg=Red

" " Mappings " let mapleader = ","

"escape insert mode inoremap jk

"save and quit noremap w :wall noremap q :q noremap qq :qall noremap ! :!q

"edit and source vimrc noremap sv :source $MYVIMRC noremap ev :vsplit $MYVIMRC

"NERDTree noremap tt :NERDTreeFocus noremap T :NERDTreeToggle

"easy fixing for ycm noremap ff :YcmCompleter FixIt

" moving single lines up and down noremap J ddp noremap K ddkP

"I don't know what this is"noremap J m:silent +g/\m^\s*$<CR>`:noh<CR> "noremap
<silent>K m:silent -g/^$/d:noh<CR> "noremap <silent><C-J> :set paste<CR>m`o<Esc>:set
nopaste "noremap :set pastem`O``:set nopaste

" moving between splits noremap h noremap j noremap k noremap l

"Compile Mappings set =^[[17;2~ noremap :call RunPython() noremap :call
CompileRun() noremap :call CompileViewRun()

" " Variables "

"Powerline let g:airline_powerline_fonts = 1 let g:airline#extensions#tabline#enabled = 1

"youcompleteme let g:ycm_server_python_interpreter='/usr/bin/python2.7'let
g:ycm_global_ycm_extra_conf = '~/vim/.ycm_extra_conf.py' let
g:completor_python_binary = '/usr/lib/python3.6/site-packagesk' let
g:ycm_min_num_of_chars_for_completion = 1 let
g:ycm_autoclose_preview_window_after_completion = 0 let g:ycm_clangd_binary_path =
'/home/ethan/.clangd/clang+llvm-9.0.0-aarch64-linux-gnu' let
g:ycm_add_preview_to_completeopt = 1

```

```

"make ycm compatible with ultisnips let g:ycm_key_list_select_completion = [", "] let
g:ycm_key_list_previous_completion = [", "] let g:SuperTabDefaultCompletionType = "

let g:UltiSnipsExpandTrigger="" let g:UltiSnipsJumpForwardTrigger="" let
g:UltiSnipsJumpBackwardTrigger="" let g:UltiSnipsEditSplit='vertical'

let g:UltiSnipsSnippetDirectories=[${HOME}/.vim/mySnips] let
g:UltiSnipsSnippetDir="mySnips"

"vimtex let g:tex_flavor = 'latex' let g:vimtex_view_method = 'zathura' set conceallevel=2 let
g:tex_conceal='abdmgs'

"markdown let g:vim_markdown_conceal_code_blocks = 0

"python-syntax let python_highlight_all = 1

"Functions function CompileRun() silent write silent !clear silent !make redraw! normal
silent !termite --class=pythonconsole -e $'bash -c 'make; ./run' endfunction

function CompileViewRun() silent write silent !clear !make redraw! normal silent !gnome-
terminal -x sh -c './run' endfunction

function RunPython() write

    silent !termite --class=pythonconsole -e $'bash -c \'python
~/Documents/Python_Project/main.py; read -n 1 -s -r -p \'Press any key to continue\'\'

endfunction

```